CoCoALib - Feature #51

polynomial coefficient extraction w.r.t. variable

30 Nov 2011 17:42 - John Abbott

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	100%
Category:	New Function	**Estimated time:**	0.00 hour
Target version:	CoCoALib-0.9953	**Spent time:**	9.95 hours

Description

Given a multivariate polynomial and an indet X (or set of indets?)
Produce a list(?) of "coeffs" w.r.t. X; probably should also
produce the powers of X corresponding to each "coeff".

Must decide:

- appropriate data structure for result (e.g. list, vector, iterator?)
- order in which the "coeffs" appear

Do we want a similar function for univariate polys? Useful? Sensible?

Related issues:

Related to CoCoA-5 - Feature #90: Make the new fn CoefficientsWRT available i...	**Closed**	**11 Feb 2012**
Related to CoCoALib - Feature #50: Polynomial content	**Closed**	**30 Nov 2011**
Related to CoCoALib - Feature #62: polynomial coefficient extraction w.r.t. s...	**Closed**	**13 Dec 2011**

History

#1 - 16 Mar 2012 18:01 - John Abbott

- Assignee set to John Abbott

JAA has made a first impl.
Still unsure about the interface (*i.e.* the type of the result & also the type of the parameter for specifying which indets).

#2 - 24 Apr 2012 11:33 - John Abbott

Part of this task has already been solved by #90 CoefficientsWRT.

JAA now needs a fn which converts a polynomial into a vector of coeffs.

Inputs are a polynomial and an ordered set of PPs;
it is assumed that the poly is a lin comb of the given PPs (o/w error).
output is a vector of coeffs giving the lin comb.

The simplest way to specify an ordered set of PPs is to represent them as a polynomial (whose coeffs can be ignored?) -- this does constrain the order of the PPs to be that of the PPordering.

A possibly more natural way to represent an ordered set of PPs is as a vector, but this leads to some complications (*e.g.* there may be duplicates, the PPs can appear in any order (which is tedious rather than truly problematic), we would have to check that each entry is indeed a PP in the correct poly ring) compared to representing the set as the support of a polynomial.

We need a good name for this fn. The name CoefficientsWRT would be fairly good, but could be confusing (or even lead to ambiguity). How about CoeffsWRTBasis? Other possibilities are CoeffVec or CoeffList.

#3 - 24 Apr 2012 14:52 - John Abbott

JAA thinks that the most useful form of CoeffsWRTBasis is one where the basis is an ordered set of PPs. Mathematically the basis can be any collection of (lin indep) polynomials; but computationally this is likely to be much costlier as the basis must be suitably prepared (*e.g.* triangularized or diagonalized).

It is likely that we will want an analogous function for module elements.
Perhaps the name should be chosen so that it can apply equally well to module elements.

#4 - 26 Apr 2012 16:14 - Anna Maria Bigatti

John Abbott wrote:

> We need a good name for this fn. The name CoefficientsWRT would be fairly good, but could be confusing (or even lead to ambiguity). How about CoeffsWRTBasis? Other possibilities are CoeffVec or CoeffList.

my preferred is **CoefficientsWRTBasis**, the name is long, but it reflects a function whose meaning is quite specialized.

#5 - 26 Apr 2012 16:18 - John Abbott

After talking to AMB, it seems best to "forget" the general function for the time being (until we have a genuine need for it).

The long name for the special fn is appropriate because it expects an unusual repr for the basis of PPs. The fn will also check that all coeffs in the "basis polynomial" are equal to 1 (o/w error).

Once this fn has been impl'ed and documented this task will finish.

#6 - 27 Apr 2012 14:56 - John Abbott

- % Done changed from 0 to 60

#7 - 28 Nov 2012 11:37 - Anna Maria Bigatti

- Status changed from New to Feedback

- Target version set to CoCoALib-0.9953

#8 - 11 Jan 2013 09:32 - Anna Maria Bigatti

I think that, by its current implementation, the order of the elements in the result of **CoefficientsWRT** is set (increasing on "PP" in the current ordering).

We should state in the documentation whether this is guaranteed or not guaranteed.

#9 - 18 Jan 2013 19:18 - John Abbott

- % Done changed from 60 to 100

After verbal discussion with Anna, we decided that we prefer that the "terms" in the result be in **decreasing order of PP**. JAA has modified the impl to guarantee this; there was already a simple test in anna.cocoa5.

#10 - 27 May 2013 18:19 - John Abbott

- Status changed from Feedback to Closed

This issue has been in "feedback" for 4 months, and no problems has arisen. So I'm closing it.